scientific reports

OPEN

Multi-angle quantum approximate optimization algorithm

Rebekah Herrman[1]✉, Phillip C. Lotshaw[2]✉, James Ostrowski[1], Travis S. Humble[2] & George Siopsis[3]

The quantum approximate optimization algorithm (QAOA) generates an approximate solution to combinatorial optimization problems using a variational ansatz circuit defined by parameterized layers of quantum evolution. In theory, the approximation improves with increasing ansatz depth but gate noise and circuit complexity undermine performance in practice. Here, we investigate a multi-angle ansatz for QAOA that reduces circuit depth and improves the approximation ratio by increasing the number of classical parameters. Even though the number of parameters increases, our results indicate that good parameters can be found in polynomial time for a test dataset we consider. This new ansatz gives a 33% increase in the approximation ratio for an infinite family of MaxCut instances over QAOA. The optimal performance is lower bounded by the conventional ansatz, and we present empirical results for graphs on eight vertices that one layer of the multi-angle anstaz is comparable to three layers of the traditional ansatz on MaxCut problems. Similarly, multi-angle QAOA yields a higher approximation ratio than QAOA at the same depth on a collection of MaxCut instances on fifty and one-hundred vertex graphs. Many of the optimized parameters are found to be zero, so their associated gates can be removed from the circuit, further decreasing the circuit depth. These results indicate that multi-angle QAOA requires shallower circuits to solve problems than QAOA, making it more viable for near-term intermediate-scale quantum devices.

Among several quantum algorithms implemented on noisy intermediate-scale quantum (NISQ) devices[1–12], the quantum approximate optimization algorithm (QAOA) offers an opportunity to approximately solve combinatorial optimization problems such as MaxCut, Max Independent Set, and Max k-cover[13–22]. QAOA tunes a set of classical parameters to optimize the cost function expectation value for a quantum state prepared by well-defined sequence of operators acting on a known initial state. Variations to the original algorithm include alternative operators and initial states[23–30] while purely classical aspects such as the parameter optimization and problem structure have been tested as well[31–36]. However, an outstanding concern is that practical implementations of QAOA require large numbers of qubits and deep circuits[37]. For example, a recent study has developed a systematic set of parameters that are argued to require $p = 30$ layers of QAOA to reach performance comparable to the conventional Goemans–Williamson algorithm on MaxCut[36], while another study has argued that hundreds of qubits or more are needed to compete with conventional solvers in time-to-solution[38]. Noise grows rapidly with circuit depth and affects the fidelity of the prepared quantum state so the performance that can be achieved from near-term quantum computers at these depths is questionable[39–49].

One approach to reduce the circuit depth of QAOA is to increase the number of classical parameters introduced in each layer, a variation that we term multi-angle QAOA (ma-QAOA). This approach was originally briefly introduced in[50]. Increasing the number of classical parameters allows for finer-grain control over the optimization of the cost function and the approximation ratio, which measures optimality relative to the known best solution. While introducing more classical parameters can lead to a more challenging optimization, a corresponding reduction in circuit depth preserves the critical resource of the quantum state. In addition, finding the absolute optimal angles is not necessary in order to see an improvement over QAOA.

Here, we quantify the advantages of using multiple parameters for each layer of QAOA. First, we prove that the approximation ratio converges to one as the number of iterations of ma-QAOA tends to infinity, a property that ensures the optimal solution is the most likely. We next demonstrate that one iteration of ma-QAOA gives an approximation ratio that is at least that of the approximation ratio after one iteration of QAOA. This shows that ma-QAOA performs at least as well as QAOA. We also show that ma-QAOA used to solve the MaxCut problem

[1]Department of Industrial and Systems Engineering, University of Tennessee at Knoxville, Knoxville, TN 37996, USA. [2]Quantum Computing Institute, Oak Ridge National Laboratory, Oak Ridge, TN 37830, USA. [3]Department of Physics and Astronomy, University of Tennessee at Knoxville, Knoxville, TN 37996, USA. ✉email: rherrma2@tennessee.edu; lotshawpc@ornl.gov

on star graphs achieves an approximation ratio of one after one iteration, while single-iteration QAOA tends to an approximation ratio of 0.75 as the number of vertices goes to infinity. This result gives a concrete example where ma-QAOA gives a strictly larger approximation ratio than QAOA. We simulate solving MaxCut using ma-QAOA and QAOA on all connected, non-isomorphic eight vertex graphs and compare the performance of the two ansatzes. In doing so, we find that the average approximation ratio for ma-QAOA after one iteration is larger than the average approximation ratio of QAOA after three iterations. In looking at larger, fifty and one-hundred vertex graphs, we see that ma-QAOA retains its advantage over QAOA, giving approximation ratios that are on average six percentage points higher after the first iteration.

Results

Multi-angle quantum approximate optimization algorithm.

We develop the multi-angle QAOA beginning with the standard formulation of the quantum approximate optimization algorithm (QAOA). The QAOA relies on a combination of classical parameter optimization and applying cost and mixing operators to a quantum state in order to approximately solve combinatorial optimization (CO) problems[13]. CO problems are defined by an objective function, $C(z)$, where z is a bit string of length n. Often, $C(z)$ is the sum over a collection of clauses,

$$C(z) = \sum_a C_a(z).$$

When solving these problems with QAOA, $C(z)$ is encoded into a matrix C with eigenvalues given by the classical cost values

$$C|z\rangle = C(z)|z\rangle. \tag{1}$$

QAOA requires two operators,

$$U(\gamma, C) = e^{-i\gamma C}$$

and

$$U(\beta, B) = e^{-i\beta B}$$

which have real-valued angle inputs $\gamma \in [0, 2\pi)$ and $\beta \in [0, \pi)$. B drives transitions between computational basis states and is typically

$$B = \sum_v B_v$$

where $B_v = \sigma_v^x$ is the Pauli-x operator acting on qubit v in the quantum system. The two operators are applied to an initial state,

$$|s\rangle = \frac{1}{\sqrt{2^n}} \sum_z |z\rangle.$$

Here the sum is over the computational basis $|z\rangle$. The QAOA ansatz operator applied p times to $|s\rangle$ is denoted p-QAOA. The state for p-QAOA is

$$|\gamma, \beta\rangle = U(\beta_p, B)U(\gamma_p, C)...U(\beta_1, B)U(\gamma_1, C)|s\rangle.$$

Since C and B are sums of matrices, we may write

$$U(\gamma, C) = e^{-i\gamma \sum_a C_a}$$

and

$$U(\beta, B) = e^{-i\beta \sum_v B_v}.$$

Instead of focusing on minimizing the classical optimization efforts in QAOA, QAOA can be modified such that it requires more classical parameters[50]. The new classical parameters are introduced to QAOA by allowing each summand of the cost and mixing operators to have its own angle instead of a single angle for the cost operator and a second angle for the mixing operator. In this modification,

$$U(\vec{\gamma}_l, C) = e^{-i\sum_a \gamma_{l,a} C_a} = \prod_a e^{-i\gamma_{l,a} C_a}$$

and

$$U(\vec{\beta}_l, B) = e^{-i\sum_v \beta_{l,v} B_v} = \prod_v e^{-i\beta_{l,v} B_v}$$

where $\vec{\gamma}_l = (\gamma_{l,a_1}, \gamma_{l,a_2}, ...)$ and $\vec{\beta}_l = (\beta_{l,v_1}, \beta_{l,v_2}, ...)$. Here, l denotes the layer, a_i denotes a specific clause, and v_j refers to a specific qubit. We call this modification multi-angle QAOA and abbreviate it ma-QAOA. Similarly to QAOA, when the operators for ma-QAOA are applied p times to the initial state, we call this p-ma-QAOA.

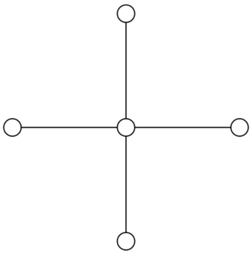

Figure 1. The star graph on five vertices.

The performance of the algorithm is typically characterized by the approximation ratio, denoted A.R.,

$$\text{A.R.} = \frac{\langle C \rangle}{C_{\max}} \tag{2}$$

which compares the expectation value of the cost operator $\langle C \rangle$ to the optimal solution value $C_{\max}$. We will write $\langle C \rangle = \langle C \rangle_p$ for p-QAOA and $\langle C \rangle = \langle C \rangle_p^{\text{ma}}$ for p-ma-QAOA.

Convergence of ma-QAOA. For QAOA, the expected value of C after p iterations is $\langle C \rangle_p = \langle \gamma, \beta | C | \gamma, \beta \rangle$. Let M_p be the maximum of $\langle C \rangle_p$ over all angles. Then, $M_p \geq M_{p-1}$. Farhi, Goldstone, and Gutmann showed that M_p tends to the maximum of the objective function, $C_{\max}$, for the CO problem being solved as p tends to infinity[13].

We similarly define the expected value of C after p iterations of ma-QAOA as $\langle C \rangle_p^{\text{ma}} = \left\langle \vec{\gamma}_{\text{ma}}, \vec{\beta}_{\text{ma}} \middle| C \middle| \vec{\gamma}_{\text{ma}}, \vec{\beta}_{\text{ma}} \right\rangle$ where $\vec{\gamma}_{\text{ma}} = (\vec{\gamma}_1, \vec{\gamma}_2, \ldots \vec{\gamma}_p)$ and $\vec{\beta}_{\text{ma}} = (\vec{\beta}_1, \vec{\beta}_2, \ldots \vec{\beta}_p)$. We also define M_p^{ma} to be the maximum of $\langle C \rangle_p^{\text{ma}}$ over all angles. Clearly, $M_p^{\text{ma}} \geq M_p$ because QAOA is the special case of ma-QAOA where $\beta_{p,u} = \beta_{p,v}$ for all $u \neq v$ and $\gamma_{p,a_i} = \gamma_{p,a_j}$ for edges $a_i \neq a_j$.

In order to show ma-QAOA gives the optimal solution to a combinatorial optimization problem, we must show $\langle C \rangle_p^{\text{ma}}$ converges to $C_{\max}$ as p tends to infinity. Convergence is the first main result of this work.

Theorem 2.1 *The multi-angle quantum approximate optimization algorithm converges to the optimal solution of a combinatorial optimization problem as* $p \to \infty$.

The proof of convergence is given in section "Methods".

MaxCut problem and performance on star graphs. In graph theory, a graph $G = (V, E)$ consists of a collection of vertices, V, and edges, E, which are pairs of vertices. MaxCut is a CO problem defined with respect to a graph. For QAOA, each qubit corresponds to a vertex in G and the cost operator is[13]

$$C = \sum_{ij \in E} \frac{1}{2}(-\sigma_i^z \sigma_j^z + 1).$$

The goal of the problem is partition the vertices into two sets such that the number of edges with endpoints in each set is maximized.

A star graph on n vertices is a graph that consists of one vertex of degree $n-1$, called the center. All other vertices of the graph have degree one, meaning each vertex is connected to the center and only the center. An example can be seen in Fig. 1. All stars are trees, and are thus bipartite, so the optimal MaxCut solution includes all edges of the graph. In order to show ma-QAOA outperforms QAOA when solving MaxCut on star graphs, we show that $\langle C \rangle_1^{\text{ma}} = 1$ and $\langle C \rangle_1$ tends to 0.75 as n tends to infinity. The proof is found in section "Methods".

Computational results. In order to test how ma-QAOA performs, we simulated the algorithm on a collection of one-hundred triangle-free 3-regular graphs with fifty vertices and one-hundred triangle-free 3-regular graphs with 100 vertices and compared the approximation ratios calculated with ma-QAOA to those of 1-QAOA. We also performed the same calculations with fifty modified $G_{n,p}$ random graphs with fifty and one-hundred vertices each; approximation ratio results for all large graphs are summarized in Table 1. In the $G_{n,p}$ model, n sets the number of vertices, and p is the probability that an edge exists. In particular, we examined $G_{50,0.08}$ and $G_{100,0.035}$ in order to create random graphs that have average degree approximately three. After randomly generating the graphs, triangles were removed by randomly removing edges from each triangle. For these sets of triangle-free graphs we can compute $\langle C \rangle_1^{\text{ma}}$ for large n using the analytical result of Theorem 4.1. Table 1 shows the average approximation ratios for each collection of graphs with ma-QAOA and 1-QAOA, as well as the changes in the approximation ratio and percent change in the approximation ratio gap. This approximation ratio gap is the percent difference between one minus the approximation ratio for 1-QAOA and one minus the approximation ratio for ma-QAOA. The ma-QAOA has a higher average approximation ratio and gives a significant percent increase in approximation ratio gap for each collection of graphs. These simulations only compare

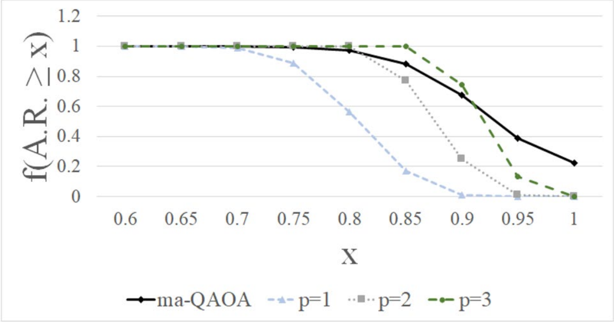

Figure 2. The fraction of non-isomorphic, eight-vertex graphs with approximation ratios at least X (f (A.R. $\geq$ X)) for 1-ma-QAOA and p-QAOA. The lines are included in order to outline the shape of each distribution.

Graph type	Average A.R. for 1-QAOA	Average A.R. for ma-QAOA	Change in A.R.	Percent change in gap (1-AR)
50 vertex 3-regular	0.7617	0.8123	0.0506	21.26%
100 vertex 3-regular	0.7562	0.8000	0.0438	17.98%
Modified $G_{50,0.08}$	0.7554	0.8156	0.0602	24.65%
Modified $G_{100,0.035}$	0.7497	0.8098	0.0602	24.04%

Table 1. The average approximation ratio (A.R.) for a collection of one-hundred 3-regular graphs with fifty vertices, one-hundred 3-regular graphs with 100 vertices, fifty modified $G_{50,0.08}$ random graphs, and fifty modified $G_{100,0.035}$ random graphs.

QAOA type	Average approximation ratio for all eight vertex graphs
ma-QAOA	0.9257
1-QAOA	0.8061
2-QAOA	0.8767
3-QAOA	0.9192

Table 2. The average approximation ratio for all connected, non-isomorphic graphs eight vertices.

1-QAOA to 1-ma-QAOA, however, the next set of computational results compares 1-ma-QAOA to p-QAOA for $p \leq 3$ on all connected, non-isomorphic graphs.

In previous work, we determined $\langle C \rangle_1$, $\langle C \rangle_2$, and $\langle C \rangle_3$ for all connected, non-isomorphic eight vertex graphs and compiled them into an online data set[35,51]. For this work, we calculated the angles that maximize $\langle C \rangle_1^{\text{ma}}$ and compared $\langle C \rangle_p$ to $\langle C \rangle_1^{\text{ma}}$. On average, the performance of ma-QAOA is comparable to 3-QAOA on these graphs. Table 2 shows that ma-QAOA has a higher average approximation ratio than 1-QAOA and 2-QAOA on all eight vertex graphs. However, the average approximation ratio for one iteration of ma-QAOA is larger than the average approximation ratio for 3-QAOA.

Figure 2 shows how the distribution of approximation ratios for ma-QAOA compares to the approximation ratios for up to three iterations of QAOA for all connected, non-isomorphic eight vertex graphs. The percentage of graphs with approximation ratio at least 0.95 is significantly higher with ma-QAOA than up to three levels of QAOA. The fraction of graphs with approximation ratio at least 0.85 and 0.9 is higher for 3-QAOA than ma-QAOA, however significantly more graphs have an approximation ratio of at least 0.95 with ma-QAOA.

Measurement reliability. We quantify the number of measurements to obtain a reliable result from ma-QAOA and QAOA using a simple noise model with Kraus-operator error channels acting after each unitary operator in the ansatz. On fully connected hardware, the numbers of one-qubit unitary operators and two-qubit unitary operators per iteration of QAOA for MaxCut equal the numbers of vertices n and edges m in the graph, respectively. On connected $n = 8$ vertex graphs, $7 \leq m \leq 28$. Following these unitary and channel operators, the circuit produces a final state $\rho = F\rho_{\text{ideal}} + (1 - F)\rho_{\text{noise}}$, where F is the probability associated with the ideal noiseless evolution component ρ_{ideal}[52]. Assuming error rates of ϵ_n and ϵ_m for each one- and two-qubit unitary respectively, $F = (1 - \epsilon_n)^{np}(1 - \epsilon_m)^{mp}$.

A measurement projects ρ onto a basis state $|z\rangle$ and the total set of measurement probabilities is described by $\rho' = \sum_z \Pi_z \rho \Pi_z$, with $\Pi_z = |z\rangle\langle z|$. The expected number of measurements to sample a result $|z\rangle$ from the ideal distribution is $1/F$ in the worst-case[48], when $\text{Tr}\rho'_{\text{ideal}}\rho'_{\text{noise}} = 0$; the number of measurements can decrease

n	Percent of v with $\beta_v = 0$	Percent of a with $\gamma_a = 0$
8	15.030	25.449
50 (3-reg.)	13.000	18.147
50 (E.R.)	11.440	14.381
100 (3-reg.)	14.690	19.973
100 (E.R.)	12.900	16.541

Table 3. The percent of β_v and γ_a, rounded to three decimal places, that are zero when optimizing ma-QAOA on the family of graphs found in the first column.

		$\epsilon_n = \epsilon_{\langle m \rangle} = 0.01$			$\epsilon_n = 0.01, \epsilon_{\langle m \rangle} = 0.05$		
n	$\langle m \rangle$	$p = 1$	$p = 2$	$p = 3$	$p = 1$	$p = 2$	$p = 3$
8	14.4	1.05	1.32	1.65	1.22	2.77	6.28
50 (3-reg.)	75	1.22	4.30	15.10	2.15	166.16	1×10^4
50 (E.R.)	87.2	1.20	4.77	18.94	2.02	291.78	4×10^4
100 (3-reg.)	150	1.57	19.32	238.39	5.39	3×10^4	2×10^8
100 (E.R.)	167.34	1.50	22.08	324.26	4.71	7×10^4	1×10^9

Table 4. The ratio of the expected number of measurements to obtain a sample from the noiseless distribution for p-QAOA relative to 1-ma-QAOA on an n vertex graph, assuming an average number of edges $\langle m \rangle$ for graphs in the datasets.

depending on the specific state and noise process, but to keep the discussion general we take the expected number of measurements as $1/F$. We compute F using the average numbers of edges $\langle m \rangle$ for graphs in our datasets, for example $\langle m \rangle = 14.4$ at $n = 8$, but note each specific graph has an integer number of edges. Assuming $p = 1$, $n = 8$, $\langle m \rangle = 14.4$, and an error rate of 1% for each unitary operator, the expected number of measurements to obtain a sample from the noiseless distribution is 1.25.

We find that parameter optimization with ma-QAOA yields angles of zero for a subset of the edge and vertex unitary operators and we use this in the calculation of F. Since $\exp(-i\gamma_{p,a}C_a) = \mathbb{I} = \exp(-i\beta_{p,v}B_v)$ when $\gamma_{p,a} = 0$ and $\beta_{p,v} = 0$, all unitary operators with an angle of zero may be excluded from the optimized circuit. This decreases the exponent of the first and second terms in F by the number of vertex and edge operators that have zero angles, respectively, and thus reduces the amount of noise in ma-QAOA relative to QAOA. Table 3 gives the percent of zero angles, rounded to three decimal places, for each collection of graphs that were studied.

Table 4 shows the ratio of the expected number of measurements needed to sample from the noiseless distribution for p-QAOA relative to ma-QAOA for each collection of graphs with varying values of $\epsilon_{\langle m \rangle}$, using the average reduction in gates for ma-QAOA from Table 3. Note that if the $\epsilon_{\langle m \rangle} = 0.05$, the number of samples increases rapidly with p.

From the performance bound of Theorem 2.1, ma-QAOA will never need more layers than standard QAOA to reach a given approximation ratio. Whenever standard QAOA requires more layers than ma-QAOA, the additional noise from these layers will lead to an increase in the number of samples that are needed according to our model. Since one iteration of ma-QAOA is comparable to three iterations of QAOA on eight vertex graphs, if the trend holds for larger graphs, ma-QAOA has the potential to require significantly fewer samples than QAOA.

Computing angles.

With a larger number of variables to optimize, the ma-QAOA method requires more classical effort to find angles that optimize the approximation ratio. However, it is not necessary to identify exact optimal angles, only to find angles that are better than QAOA angles.

We used the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm to compute angles for the 8-vertex graphs; details can be found in "Methods" section. Figure 3 shows how the approximation ratio improves on average across all iterations of BFGS for each ansatz studied for a random sample of eight vertex graphs. Note that after approximately ten iterations, ma-QAOA tends to achieve a higher approximation ratio than any of the p-QAOA. We do note that the time required to perform each iteration of BFGS is slower for ma-QAOA, as the number of gradient components is linearly dependent on the number of variables being optimized.

Scaling.

We assess the scalability of ma-QAOA using computed optimized $\langle C \rangle$ for sets of triangle-free Erdős-Rényi and 3-regular graphs with $n = 50$ and $n = 100$ vertices. The computational details are given in section "Methods". We compare the run times for typical graph optimizations to assess how the ma-QAOA parameter optimization time increases with graph size.

For the Erdős-Rényi graphs, the time for a single optimization for $n = 50$ was 0.10 seconds, for $n = 100$ it was 0.46 seconds. We attribute the difference primarily to the scaling in the calculation of the gradient, which is the most expensive calculation in the optimization. Our approach computes each of the $n + m$ derivatives

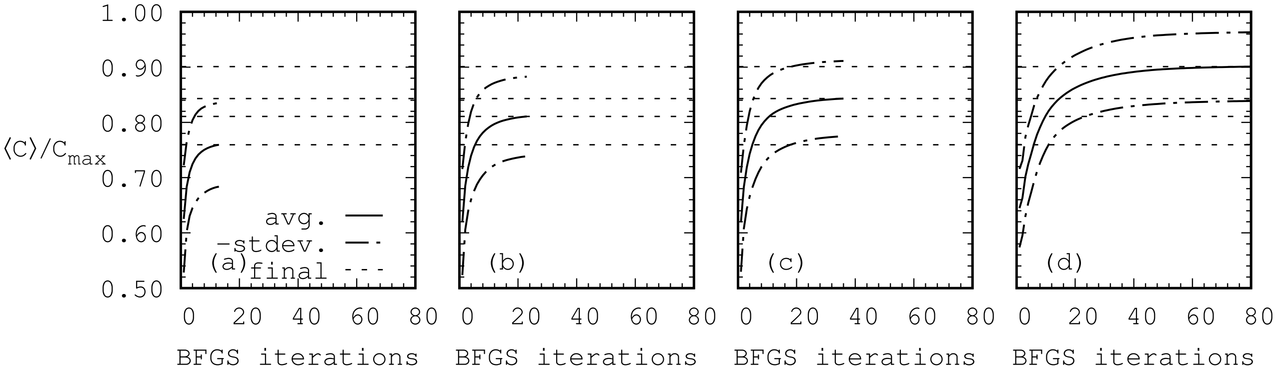

Figure 3. Typical behavior of the BFGS search algorithm for 100 random eight vertex graphs each with 100 random seeds in BFGS, for regular QAOA at (**a**) $p = 1$, (**b**) $p = 2$, (**c**) $p = 3$ and for (**d**) ma-QAOA at $p = 1$. Solid curves show averages over graphs and seeds, dashed-dotted lines show the average ± standard deviation and dotted lines show the final values. Each curve terminates at the average final iteration of the BFGS algorithm for the dataset.

$\partial\langle C_{p,uv}\rangle^{\text{ma}}/\partial\beta_{p,w}$ and $\partial\langle C_{p,uv}\rangle^{\text{ma}}/\partial\gamma_{p,jk}$ for each of the m terms $\langle C_{p,uv}\rangle^{\text{ma}}$ in the cost function, giving a total number of terms $\sim (n+m)m$. The time to compute each term will vary with the degree of the graph, as this determines the number of cosine terms in Theorem 4.1; however, for our graphs the degree is approximately constant hence can be neglected in the scaling. For our graphs $m \sim n$ on average, so the overall scaling is $\sim n^2$, which is consistent with the $\approx 4\times$ increase in time when n is doubled from $n = 50$ to $n = 100$. We attribute the remainder of the time difference to variations in the number of iterations as n and m increase.

It is interesting to consider scaling of the optimization time with the number of vertices n for instances beyond the current dataset. For a gradient-based optimization this requires computing $\partial\langle C^{\text{ma}}\rangle/\partial\theta = \sum_a \partial\langle C_a\rangle/\partial\theta$ for each parameter θ, for each step in the optimization. For MaxCut and a variety of other problems[53], the number of clauses C_a is poly(n), and so there are poly(n) parameters and poly(n) partial derivatives $\partial\langle C_a\rangle/\partial\theta$ in the gradient. There are situations in which the time to compute each $\partial\langle C_a\rangle/\partial\theta$ is independent of n, specifically, when p and the graph structure are fixed such that each partial derivative can be computed using n-independent "sub-graphs"[13]. Then we need to compute poly(n) terms with fixed compute time per term, so the overall time to compute the gradient scales as poly(n). The gradient based optimization approach BFGS exhibits super-linear convergence on a variety of practical problems[54], which supports the idea that the number of steps will not scale problematically with n. Perhaps counterintuitively, a recent investigation of variational quantum algorithms suggests that algorithms with more parameters have fewer local optima and achieve better convergence to global optima[55], suggesting ma-QAOA may require fewer BFGS step to optimize than standard QAOA.

Discussion

We have shown that multi-angle QAOA converges to an optimal solution, and furthermore that $\langle C\rangle_1^{\text{ma}} \geq \langle C\rangle_1$, as QAOA is a special case of ma-QAOA. Additionally, the analysis of star graphs shows that there is a family of graphs that always gives larger $\langle C\rangle$ for MaxCut when solved with ma-QAOA than when solved with QAOA. We find significant increases in the approximation ratio in numerical optimizations for large triangle-free graphs and over the set of all non-isomorphic graphs with eight vertices, hence fewer layers are required to reach the same performance as QAOA. We also show that optimized rotation angles are often zero in ma-QAOA and this reduces the number of unitary operators per layer relative to QAOA. In the presence of noise, the reduction in number of layers and in the number of unitary operators per layer can significantly decrease the expected number of measurements needed to sample a result $|z\rangle$ in the distribution of the noiseless state. This could be a significant advantage for computations on noisy quantum hardware.

Interestingly, some graphs do not have a significantly higher $\langle C\rangle$ when solving MaxCut with ma-QAOA versus QAOA. It would be useful to characterize for which graphs the increase in $\langle C\rangle$ from QAOA to ma-QAOA is insignificant. This would help determine the appropriate ansatz to use in order to solve MaxCut on the graph.

One drawback to ma-QAOA is that the number of classically optimized parameters is $n + m$ per layer, where n is the number of vertices of G and m is the number of edges. An argument can be made that if x parameters are required to optimize one iteration of ma-QAOA, the results should be compared to QAOA with the same number of parameters. This approach would require $p \approx \frac{x}{2}$ iterations of QAOA, which may not be feasible on current hardware as a large number of layers will accumulate considerable noise. From this perspective, it is advantageous to incorporate additional parameters into a small number of circuit layers. It could be interesting to consider the comparison with the same numbers of parameters from a theoretical perspective, but it is beyond our scope here.

From a practical standpoint, one way to solve optimal ma-QAOA angles would be to calculate β and γ that optimize QAOA. We can use those angles as the initial point of a BFGS search for the optimal $\beta_{p,v}$ and γ_{p,a_i} for all vertices v and edges a_i. Overall, however, the results seem to indicate that good parameters can be found in polynomial time. As many combinatorial optimization problems, like MaxCut, are NP-Hard, any polynomially-bounded effort that improves performance is likely to improve performance at large scale.

Methods

Proof of convergence.

Proof Recall that QAOA converges to the optimal solution for a combinatorial optimization problem, which is the maximum over the objective function[13]. Thus, in order to show convergence of ma-QAOA, we need only bound ma-QAOA from below by the value of QAOA. However, it is clear that the optimal expected value of the cost function for ma-QAOA can be no lower than that of QAOA, since QAOA is a special case of ma-QAOA when all $\gamma_{p,ij} = \gamma_{p,kl}$ and all $\beta_{p,a} = \beta_{p,b}$ for all edges ij, kl and all vertices a, b. □

Formula for $\langle C \rangle$.

In order to prove that $\langle C\rangle_1^{\mathrm{ma}} = 1$ for MaxCut on star graphs, we derive a formula that calculates $\langle C\rangle_1^{\mathrm{ma}}$ for MaxCut on triangle-free graphs.

Theorem 4.1 *Let $\beta'_{p,u} = 2\beta_{p,u}$ and $\beta'_{p,v} = 2\beta_{p,v}$ The expected value of C after one iteration of ma-QAOA applied to MaxCut for triangle-free graphs G is*

$$\left\langle \vec{\gamma}_1\vec{\beta}_1 \middle| C_{uv} \middle| \vec{\gamma}_1\vec{\beta}_1 \right\rangle = \frac{1}{2} + \frac{1}{2}\sin\gamma_{1,uv}(\cos\beta'_{1,v}\sin\beta'_{1,u}\prod_w \cos\gamma_{1,uw} + \cos\beta'_{1,u}\sin\beta'_{1,v}\prod_x\cos\gamma_{1,vx})$$

where $w \in Nbhd(u) \setminus v$ and $x \in Nbhd(v) \setminus u$.

The neighborhood of a vertex x, denoted $Nbhd(x)$, is the set of vertices y such that $xy \in E(G)$.

Proof The proof of this result relies on the Pauli-solver algorithm, which is explained in detail in[56]. The proof of the result is virtually identical to that for QAOA on triangle-free graphs, but we include the proof here for completeness.

Consider edge uv and consider acting on $C_{uv} = (1/2)(\mathbb{I} - Z_uZ_v)$ by conjugation of the mixing operator, $\prod_{i\in V} e^{-i\beta_{1,i}B_i}$, followed by conjugation of the phase operator, $\prod_{uv\in E} e^{-i\gamma_{1,uv}C_{uv}}$. We have that

$$\begin{aligned}\prod_{i\in V} e^{i\beta_{1,i}B_i} Z_uZ_v \prod_{i\in V} e^{-i\beta_{1,i}B_i} &= e^{2i\beta_{1,u}X_u} e^{2i\beta_{1,v}X_v} Z_uZ_v \\ &= \cos 2\beta_{1,u}\cos 2\beta_{1,v}Z_uZ_v + \cos 2\beta_{1,v}\sin 2\beta_{1,u}Y_vZ_u \\ &\quad + \cos 2\beta_{1,u}\sin 2\beta_{1,v}Z_vY_u + \sin 2\beta_{1,u}\sin 2\beta_{1,v}Y_uY_v.\end{aligned} \tag{3}$$

Note that the first term commutes with $\prod_{uv\in E} e^{-i\gamma_{1,uv}C_{uv}}$, so does not contribute to the expected value. Let V_u be the neighborhood of u in $V(G)$. Conjugating the third term of Eqn. (3) by $\prod_{uv\in E} e^{-i\gamma_{1,uv}C_{uv}}$, we get

$$\begin{aligned}\langle s|\Upsilon^\dagger Y_uZ_v\Upsilon|s\rangle &= \langle s|e^{2i\gamma_{1,uv}C_{uv}} e^{2i\sum_{a\in V_u\setminus v}\gamma_{1,ua}C_{ua}} Y_uZ_v|s\rangle \\ &= \langle s|e^{-i\gamma_{1,uv}Z_uZ_v} e^{-i\sum_{a\in V_u\setminus v}\gamma_{1,ua}Z_uZ_a} Y_uZ_v|s\rangle \\ &= \langle s|(\mathbb{I}\cos\gamma_{1,uv} - i\sin\gamma_{1,uv})Z_uZ_v \\ &\quad \prod_{a\in V_u}(\mathbb{I}\cos\gamma_{1,ua} - i\sin\gamma_{1,ua}Z_uZ_a)Y_uZ_v|s\rangle \\ &= -\sin\gamma_{1,uv}\prod_{a\in V_u}\cos\gamma_{1,ua},\end{aligned}$$

where $\Upsilon = e^{-i\gamma_{1,uv}C_{uv}} e^{-i\sum_{a\in V_u\setminus v}\gamma_{1,ua}C_{ua}}$, and $\Upsilon^\dagger$ is its Hermitian conjugate. By symmetry, the term for Z_uY_v is $-\sin\gamma_{1,uv}\prod_{b\in V_v\setminus u}\cos\gamma_{1,vb}$, where V_v is the neighborhood of v in V. Factoring in the coefficient $-1/2$ of Z_uZ_v in C_{uv} gives the final two terms in the theorem.

Now, let us conjugate the last term of Eq. (3). Doing so, we get

$$\begin{aligned}&\langle s|e^{i\sum_{gh\in E}\gamma_{1,gh}C_{gh}} Y_uY_v e^{-i\sum_{gh\in E}\gamma_{1,gh}C_{gh}}|s\rangle \\ &= \langle s|\prod_{a\in V_u\setminus v}(\mathbb{I}\cos\gamma_{1,ua} - i\sin\gamma_{1,ua}Z_uZ_a) \times \prod_{b\in V_v\setminus u}(\mathbb{I}\cos\gamma_{1,vb} - i\sin\gamma_{1,vb}Z_vZ_b)Y_uY_v|s\rangle\end{aligned}$$

The simplest terms that contribute to the expected value are of the form

$$\sin\gamma_{1,uc}\sin\gamma_{1,vc}\prod_{x\neq y}\cos\gamma_{1,ux}\cos\gamma_{1,vy}$$

and there are f of these where f is the number of triangles containing uv. The higher order terms only contribute to the expected value if there are triangles in the graph. Thus, the last term of Eqn. (3) contributes nothing to the expected value of triangle-free graphs.

Combining these expressions gives the theorem. □

Star graphs. First, we will show that $\langle C_{ij} \rangle$ approaches 0.75 as n tends to infinity for QAOA. Since there are $n-1$ edges in a star on n vertices, this implies $\langle C \rangle$ tends to $0.75(n-1)$. Additionally, $n-1$ is the size of the optimal MaxCut solution, so $\langle C \rangle_1 / C_{\max} = 0.75$.

Wang, Hadfield, Jiang, and Rieffel showed that[57]

$$\langle C_{ij} \rangle_1 = \frac{1}{2} + \frac{1}{4}(\sin 4\beta \sin \gamma)(\cos^d \gamma + \cos^e \gamma) - \frac{1}{4}(\sin^2 2\beta \cos^{d+e-f} \gamma)(1 - \cos^f 2\gamma) \tag{4}$$

where d is the $\deg(i) - 1$, e is the $\deg(j) - 1$ and f is the number of triangles containing edge ij[56,57].

Let us consider the above formula applied to a star graph. Without loss of generality, let j be the center of each star. Then $d = 0$, $e = n-2$, and $f = 0$, since star graphs are trees. For each edge of the star, Eq. (4) reduces to

$$\langle C_{ij} \rangle_1 = \frac{1}{2} + \frac{1}{4}(\sin 4\beta \sin \gamma)(1 + \cos^{n-2} \gamma).$$

We set $\beta = \pi/8$, which implies $\sin 4\beta = 1$, since only one trigonometric function has β as an argument. As n tends to infinity, note $\cos^{n-2}\gamma$ tends to zero unless $\gamma = k\pi$ for some $k \in \mathbf{N}$. However, if $\gamma = k\pi$, $\sin \gamma = 0$. Thus, this quantity is maximized when $\gamma \neq k\pi$, which implies $\langle C_{ij} \rangle_1$ tends to 0.75 for star graphs.

In order to prove $\langle C \rangle^{\mathrm{ma}} = n - 1$ for ma-QAOA on star graphs, we examine Theorem 4.1. Without loss of generality, let u be a leaf vertex and v be the center. Note that the first product is empty, since the leaf vertices have no neighbors except the center. Thus, Theorem 4.1 reduces to

$$\left\langle \vec{\gamma}_1 \vec{\beta}_1 \middle| C_{uv} \middle| \vec{\gamma}_1 \vec{\beta}_1 \right\rangle = \frac{1}{2} + \frac{1}{2} \sin \gamma_{1,uv} (\cos \beta'_{1,v} \sin \beta'_{1,u} + \cos \beta'_{1,u} \sin \beta'_{1,v} \prod_x \cos \gamma_{1,vx})$$

Now, recall $\left\langle \vec{\gamma}_1 \vec{\beta}_1 \middle| C_{uv} \middle| \vec{\gamma}_1 \vec{\beta}_1 \right\rangle \leq 1$, as two vertices that have an edge between them add one to the objective function if they are in different sets. In order to obtain equality, we can set $\gamma_{1,uv} = \pi/2$, as it is an argument for only a single sine term. Next, note that if either term in the parenthesis is one, the other must be zero. Also, setting one term equal to one allows gives an expected value of one for the edge. Let $\beta'_{1,u} = \pi/2$ and $\beta'_{1,v} = 0$. Then $\cos \beta'_{1,v} = \sin \beta'_{1,u} = 1$ while $\cos \beta'_{1,u} = \sin \beta'_{1,v} = 0$. Thus, the first term in the parenthesis is one and the second is zero. This allows us to set $\gamma_{1,vx} = \pi/2$ for all $x \in Nbhd(v)$. Since each of the $n-1$ edges in the star are described similarly, $\langle C \rangle_1^{\mathrm{ma}} = n - 1$ for all n. The size of the optimal cut on a star graph is $n-1$, so $\langle C \rangle_1^{\mathrm{ma}} / C_{\max} = 1$.

Setup for computational results. In order to calculate the angles that maximize $\langle C \rangle_p$ and $\langle C \rangle_1^{\mathrm{ma}}$ for the eight vertex graphs, we used the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm[58]. The algorithm inputs an initial collection of angles and then uses a numerical gradient and second order approximate Hessian to find angles that converge to local maxima of $\langle C \rangle_p$ and $\langle C \rangle_1^{\mathrm{ma}}$. For the eight vertex graphs, one-hundred random seeds were used to optimize $\langle C \rangle_1^{\mathrm{ma}}$. The results for the $\langle C \rangle_p$ were taken from the online dataset[51] of Ref.[35], where we performed an exhaustive analysis of QAOA performance on small graphs. These used fifty seeds for $p = 1$, one-hundred seeds for $p = 2$, and one-thousand seeds for $p = 3$.

For the fifty and one-hundred vertex graphs, we used the method of moving asymptotes (MMA) algorithm[59,60], but note that calculations with BFGS gave similar results. The $\langle C \rangle_1$ were computed using Eq. (4) and the reported results were taken as the best from one-thousand initial seeds in MMA optimizations. The $\langle C \rangle_1^{\mathrm{ma}}$ were computed with Theorem 4.1 and MMA optimization. We report results as the best optimized values from one-thousand seeds at $n = 50$ and from one-hundred seeds at $n = 100$.

Data availability

The datasets generated during and/or analysed during the current study are available in the Multi-Angle-QAOA repository, https://code.ornl.gov/5ci/multi-angle-qaoa.

Received: 25 January 2022; Accepted: 8 April 2022
Published online: 26 April 2022

References

1. Cruz, D. *et al.* Efficient quantum algorithms for GHZ and W states, and implementation on the IBM quantum computer. *Adv. Quant. Technol.* **2**(5–6), 1900015 (2019).
2. Zhang, J., Hegde, S. S. & Suter, D. Efficient implementation of a quantum algorithm in a single nitrogen-vacancy center of diamond. *Phys. Rev. Lett.* **125**(3), 030501 (2020).
3. Godfrin, C. *et al.* Operating quantum states in single magnetic molecules: Implementation of Grover's quantum algorithm. *Phys. Rev. Lett.* **119**(18), 187702 (2017).
4. Borle, A., Elfving, V. & Lomonaco, S. J. Quantum approximate optimization for hard problems in linear algebra. *SciPost Phys. Core* **4**(4), 031 (2021).
5. Karamlou, A. H. *et al.* Analyzing the performance of variational quantum factoring on a superconducting quantum processor. *NPJ Quant. Inf.* **7**(1), 1–6 (2021).
6. Hempel, C. *et al.* Quantum chemistry calculations on a trapped-ion quantum simulator. *Phys. Rev. X* **8**(3), 031022 (2018).
7. Huerta Alderete, C. *et al.* Quantum walks and Dirac cellular automata on a programmable trapped-ion quantum computer. *Nat. Commun.* **11**(1), 1–7 (2020).
8. Xue, P., Sanders, B. C. & Leibfried, D. Quantum walk on a line for a trapped ion. *Phys. Rev. Lett.* **103**(18), 183602 (2009).
9. Linke, N. M. *et al.* Experimental comparison of two quantum computing architectures. *Proc. Natl. Acad. Sci. USA* **114**(13), 3305–3310 (2017).

10. Pagano, G. *et al.* Quantum approximate optimization of the long-range Ising model with a trapped-ion quantum simulator. *Proc. Natl. Acad. Sci. USA* **117**(41), 25396–25401 (2020).
11. Bengtsson, A. *et al.* Quantum approximate optimization of the exact-cover problem on a superconducting quantum processor. arXiv:1912.10495 (2019).
12. Harrigan, M. P. *et al.* Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. *Nat. Phys.* **17**(3), 332–336 (2021).
13. Farhi, E., Goldstone, J., & Gutmann, S. A quantum approximate optimization algorithm. arXiv:1411.4028 (2014).
14. Saleem, Z. H. Max-independent set and the quantum alternating operator ansatz. *Int. J. Quant. Inf.* **18**(04), 2050011 (2020).
15. Cook, J., Eidenbenz, S., & Bärtschi, A. The quantum alternating operator ansatz on maximum k-vertex cover. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 83–92 (2020).
16. Crooks, G. E. Performance of the quantum approximate optimization algorithm on the maximum cut problem. arXiv:1811.08419 (2018).
17. Pichler, H.s, Wang, S.-T., Zhou, L., Choi, S., & Lukin, M. D. Quantum optimization for maximum independent set using rydberg atom arrays. arXiv:1808.10816 (2018).
18. Farhi, E., Gamarnik, D., & Gutmann, S. The quantum approximate optimization algorithm needs to see the whole graph: Worst case examples. arXiv:2005.08747 (2020).
19. Farhi, E., Gamarnik, D., & Gutmann, S. The quantum approximate optimization algorithm needs to see the whole graph: A typical case. arXiv preprint arXiv:2004.09002 (2020).
20. Wurtz, J. & Love, P. MaxCut quantum approximate optimization algorithm performance guarantees for $p > 1$. *Phys. Rev. A* **103**, 042612 (2021).
21. Ward, J., Otterbach, J., Crooks, G., Rubin, N. & da Silva, M. QAOA performance benchmarks using Max-Cut. *APS March Meet. Abstracts* **2018**, R15-007 (2018).
22. Shaydulin, R., & Alexeev, Y. Evaluating quantum approximate optimization algorithm: A case study. In *2019 tenth international green and sustainable computing conference (IGSC)*, 1–6. (IEEE, 2019).
23. Bärtschi, A., & Eidenbenz, S. Grover mixers for QAOA: Shifting complexity from mixer design to state preparation. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 72–82. (IEEE, 2020).
24. Hadfield, S. *et al.* From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms* **12**(2), 34 (2019).
25. Wurtz, J. & Love, P. J. Classically optimal variational quantum algorithms. *IEEE Trans. Quant. Eng.* **2**, 1–7 (2021).
26. Egger, D. J., Mareček, J. & Woerner, S. Warm-starting quantum optimization. *Quantum* **5**, 479 (2021).
27. Zhu, L., Tang, H. L., Barron, G., Calderon-Vargas, F., Mayhall, N., Barnes, E., & Economou, S. Improving the performance of quantum approximate optimization algorithm through an adaptive, problem-tailored ansatz. arXiv:2005.10258 (2020).
28. Tate, R., Farhadi, M., Herold, C., Mohler, G., & Gupta, S. Bridging classical and quantum with SDP initialized warm-starts for QAOA. arXiv:2010.14021 (2020).
29. Wang, Z., Rubin, N. C., Dominy, J. M. & Rieffel, E. G. *XY* mixers: Analytical and numerical results for the quantum alternating operator ansatz. *Phys. Rev. A* **101**, 012320 (2020).
30. Golden, J., Bärtschi, A., O'Malley, D., & Eidenbenz, S. Threshold-based quantum optimization. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 137–147. (IEEE, 2021).
31. Herrman, R. *et al.* Impact of graph structures for QAOA on MaxCut. *Quant. Inf. Process.* **20**, 1–10 (2021).
32. Shaydulin, R., Hadfield, S., Hogg, T. & Safro, I. Classical symmetries and the quantum approximate optimization algorithm. *Quant. Inf. Process.* **20**, 1–10 (2020).
33. Khairy, S., Shaydulin, R., Cincio, L., Alexeev, Y. & Balaprakash, P. Reinforcement learning for quantum approximate optimization. *Supercomputing* **19**, 1–10 (2019).
34. Wauters, M. M., Panizon, E., Mbeng, G. B. & Santoro, G. E. Reinforcement-learning-assisted quantum optimization. *Phys. Rev. Res.* **2**(3), 033446 (2020).
35. Lotshaw, P. C., Humble, T. S., Herrman, R., Ostrowski, J. & Siopsis, G. Empirical performance bounds for quantum approximate optimization. *Quant. Inf. Process.* **20**, 403 (2021).
36. Wurtz, J. & Lykov, D. Fixed-angle conjectures for the quantum approximate optimization algorithm on regular MaxCut graphs. *Phys. Rev. A* **104**, 052419 (2021).
37. Herrman, R., Ostrowski, J., Humble, T. S. & Siopsis, G. Lower bounds on circuit depth of the quantum approximate optimization algorithm. *Quant. Inf. Process.* **20**(2), 1–17 (2021).
38. Guerreschi, G. G. & Matsuura, A. Y. QAOA for Max-Cut requires hundreds of qubits for quantum speed-up. *Sci. Rep.* **9**, 1–10 (2019).
39. Xue, C., Chen, Z.-Y., Yu-Chun, W. & Guo, G.-P. Effects of quantum noise on quantum approximate optimization algorithm. *Chin. Phys. Lett.* **38**(3), 030302 (2021).
40. Wang, S. *et al.* Noise-induced barren plateaus in variational quantum algorithms. *Nat. Commun.* **12**(1), 1–11 (2021).
41. Murali, P., Baker, J. M. Javadi-Abhari, A., Chong, F. T. & Martonosi, M. Noise-adaptive compiler mappings for noisy intermediate-scale quantum computers. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, 1015–1029 (2019).
42. Sun, J. *et al.* Mitigating realistic noise in practical noisy intermediate-scale quantum devices. *Phys. Rev. Appl.* **15**(3), 034026 (2021).
43. Marshall, J., Wudarski, F., Hadfield, S. & Hogg, T. Characterizing local noise in QAOA circuits. *IOP Sci. Notes* **1**(2), 025208 (2020).
44. Alam, M., Ash-Saki, A., & Ghosh, S. Design-space exploration of quantum approximate optimization algorithm under noise. In *2020 IEEE Custom Integrated Circuits Conference (CICC)*, 1–4. (IEEE, 2020).
45. Alam, M., Ash-Saki, A., & Ghosh, S. Analysis of quantum approximate optimization algorithm under realistic noise in superconducting qubits. arXiv:1907.09631 (2019).
46. Streif, M., Leib, M., Wudarski, F., Rieffel, E. & Wang, Z. Quantum algorithms with local particle-number conservation: Noise effects and error correction. *Phys. Rev. A* **103**(4), 042412 (2021).
47. Maciejewski, F. B., Baccari, F., Zimborás, Z. & Oszmaniec, M. Modeling and mitigation of cross-talk effects in readout noise with applications to the Quantum Approximate Optimization Algorithm. *Quantum* **5**, 464 (2021).
48. Lotshaw, P. C., Nguyen, T., Santana, A., McCaskey, A., Herrman, R., Ostrowski, J., Siopsis, G., & Humble, T. S.. Scaling quantum approximate optimization on near-term hardware. arXiv:2201.02247 (2022).
49. Zeng, J. *et al.* Simulating noisy variational quantum eigensolver with local noise models. *Quant. Eng.* **3**, 1–10 (2021).
50. Farhi, E., Goldstone, J., Gutmann, S., & Neven, H. Quantum algorithms for fixed qubit architectures. arXiv:1703.06199 (2017).
51. Lotshaw, P. C., & Humble, T. S.. QAOA dataset. https://code.ornl.gov/qci/qaoa-dataset-version1.
52. Koczor, B. The dominant eigenvector of a noisy quantum state. arXiv:2104.00608 (2021).
53. Lucas, A. Ising formulations of many np problems. *Front. Phys.* **2**, 1–10 (2014).
54. Wright, S. *et al.* Numerical optimization. *Science* **35**(67–68), 7 (1999).
55. Larocca, M., Ju, N., García-Martín, D., Coles, P. J., & Cerezo, M. Theory of overparametrization in quantum neural networks. arXiv:2109.11676 (2021).
56. Hadfield, S. Quantum algorithms for scientific computing and approximate optimization. arXiv:1805.03265 (2018).

57. Wang, Z., Hadfield, S., Jiang, Z. & Rieffel, E. G. Quantum approximate optimization algorithm for MaxCut: A fermionic view. *Phys. Rev. A* **97**(2), 022304 (2018).
58. Press, W. H., Flannery, B. P. & Teukolsky, S. A. *Numerical Recipes in Fortran 77: The Art of Scientific Computing* 2nd edn. (Cambridge University Press, 1993).
59. Johnson, S. G. The nlopt nonlinear-optimization package. http://github.com/stevengj/nlopt.
60. Svanberg, K. A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM J. Optim.* **12**, 555–573 (2002).

Acknowledgements

This work was supported by DARPA ONISQ program under award W911NF-20-2-0051. J. Ostrowski acknowledges the Air Force Office of Scientific Research award, AF-FA9550-19-1-0147. G. Siopsis acknowledges the Army Research Office award W911NF-19-1-0397. J. Ostrowski and G. Siopsis acknowledge the National Science Foundation award OMA-1937008. This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan. (http://energy.gov/downloads/doe-public-access-plan).

Author contributions

All authors contributed to the work enclosed in the paper and the writing of the paper.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to R.H. or P.C.L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.